PanDA/Pegasus integration: a case study

Pavlo Svirin
Problem

- In order to perform analysis users submit set of tasks
- It is possible to specify independent tasks and tasks with one parent task
- How to specify complex workflows with more than one parents?
## Analysis tasks

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Jobset</th>
<th>Type</th>
<th>Working Group</th>
<th>User</th>
<th>Destination</th>
<th>Task status</th>
<th>Nevents</th>
<th>used</th>
<th>Expected</th>
<th>Total</th>
<th>done</th>
<th>failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>19730283</td>
<td>10067</td>
<td>analy</td>
<td></td>
<td>Pavlo Svirin</td>
<td></td>
<td>done</td>
<td>1</td>
<td>1</td>
<td>(100%)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Job parameters

```
- --containerImage docker://gitlab-registry.cern.ch/hepimages/public/gpu-basic-test
- -p
- -r

log template: value="$[LOG0]" container="user.psvirin.outest001.log" dataset="user.psvirin.outest001.log"

python%20/test-gpu.py
```

### Prosys task parameters

<table>
<thead>
<tr>
<th>allowInputLAN</th>
<th>use</th>
</tr>
</thead>
<tbody>
<tr>
<td>architecture</td>
<td>@centos7</td>
</tr>
<tr>
<td>cliParams</td>
<td>prun --exec &quot;python /test-gpu.py&quot; --containerImage=docker://gitlab-registry.cern.ch/hepimages/public/gpu-basic-test --site=ANALYZE_OU_OSECRI_GPU_TEST --noBuild --outDS=user.psvirin.outest001</td>
</tr>
</tbody>
</table>
Pegasus WFMS

Developed since 2001 by ISI at the University of Southern California (USC)

Portability across heterogeneous infrastructure
  Separation of workflow description and execution
  Support for campus and leadership class clusters, OSG, XSEDE, academic and commercial clouds
  Can interact with a number of different storage systems (with different protocols)

Supports data reuse – useful in collaborations and ensemble workflow runs

Reliability
  Recovers from failures, retry, workflow-level checkpointing

Scalability
  O(million) task, O(TB) data in a workflow

Restructures workflow for performance

Support reproducibility

Web-based monitoring and debugging tools

Can be included in various user-facing infrastructures
  Graphical composition tools
  Portals, HUBZero
Pegasus architecture
Abstract Workflows (DAX)

Workflow submission also possible through Jupyter notebooks
Pegasus and PanDA integration

Workflow description

Pegasus

DAGMan description

DAGMan

Condor

Monitor

Schedd

PanDA Server
Pegasus and PanDA

- Integration with PanDA: done in December 2018
- Tested with separate jobs
- Support for tasks added in October, 2019
Pegasus: “personal” setup

• Users can run and control workflows
  • from local machine (Pegasus and Condor installation needed)
    • good if input data is on local machine
  • from a virtual machine in CERN OpenStack (an image containing pre-installed Condor and Pegasus can be created)
• user scripts can be created so they do not need to study DAX, just specify a workflow in DOT format
• Problem: proxy expiration
DOT workflow

digraph graphname {
    a -> b -> c;
b -> d;
}

a: prun ...
b: prun ...
c:
d: ...
Pegasus as a service

- A portal which can be accessed via CERN SSO
- Users upload DOT files which converted into DAX
- Proxy substitution:
  - to submit tasks under “Workflow Robot” proxy
- Advantage:
  - nothing is needed to install
  - workflow visualisation
  - can run very long workflows without manual intrusion
- Problems: how to transfer local data to dataset?
  - can be solved if data is stored somewhere on AFS
Job management: UI sketch

UI: by Ivan Tertychnyy, NRC KI
DAG visualisation: by Ross Kirsling

http://bl.ocks.org/rkirsling/5001347
Conclusions and future work

• It is possible to build complex user workflows consisting of tasks and separate jobs

• Todo and possible ways:

  • gather use-cases from users

  • implement easy client tools for fast workflow generation

  • create a pre-configured VM image with pre-installed Pegasus/Condor software

  • estimate opportunities for a workflow portal